# Artificial Intelligence in Music Education: A Critical Review

Book Section

How to cite:

Holland, Simon (2000). Artificial Intelligence in Music Education: A Critical Review. In: Miranda, Eduardo Reck ed. Readings in Music and Artificial Intelligence. Contemporary Music Studies. Routledge, pp. 239–274.



Link(s) to article on publisher's website:
http://dx.doi.org/doi:10.4324/9780203059746

# Artificial Intelligence in music education: a critical review

Simon Holland
*Department of Computing, The Open University,*
*Milton Keynes, Great Britain.*
*s.holland@open.ac.uk*

**This paper appeared as:**
Holland, S. (2000) Artificial Intelligence in Music Education: a critical review. In Miranda, E. (ed.) *Readings in Music and Artificial Intelligence*, Contemporary Music Studies Vol. 20. Harwood Academic Publishers, The Netherlands.

## Abstract

This paper reviews the principal approaches to using Artificial Intelligence in Music Education. Music is a challenging domain for Artificial Intelligence in Education (AI-ED) because music is, in general, an open-ended domain demanding creativity and problem-seeking on the part of learners and teachers. In addition, Artificial Intelligence theories of music are far from complete, and music education typically emphasises factors other than the communication of 'knowledge' to students. This paper reviews critically some of the principal problems and possibilities in a variety of AI-ED approaches to music education. Approaches considered include: Intelligent Tutoring Systems for Music; Music Logo Systems; Cognitive Support Frameworks that employ models of creativity; highly interactive interfaces that employ AI theories; AI-based music tools; and systems to support negotiation and reflection. A wide variety of existing music AI-ED systems are used to illustrate the key issues, techniques and methods associated with these approaches to AI-ED in Music.

Key words: artificial intelligence, education, music, creativity, intelligent tutoring system, human computer interaction

**Artificial Intelligence in music education: a critical review**

Simon Holland
*Department of Computing, The Open University,*
*Milton Keynes, Great Britain.*
*s.holland@open.ac.uk*

# 1. Introduction

This paper critically reviews some of the main approaches to using Artificial Intelligence in music education. The field is highly interdisciplinary, involving substantial contributions from the fields of music, education, artificial intelligence (AI), cognitive psychology, the psychology of music, social psychology, anthropology, philosophy, linguistics, human computer interaction, mathematics, computer science and many other fields. AI in Education itself is a very diverse field, dating from about 1970 (Carbonell, 1970), and it has its own developed methodologies, techniques and traditions. Rather than attempting to be comprehensive, this paper takes a representative approach. For brevity, we will abbreviate AI in Education to AI-ED, following a standard convention.

## 1.1 Definitions

The scope of AI in Education (AI-ED) is not clear-cut, so it is useful to consider definitions. One common way of defining AI in Education is: *any application of AI techniques or methodologies to educational systems*. Other definitions focus more narrowly, for example: *any computer-based learning system which has some degree of autonomous decision-making with respect to some aspect of its interaction with its users* (Self, 1995). This second definition stresses the requirement that AI techniques are used to reason at the point of interaction with the user. The reasoning might be about the subject being taught, about the best teaching approach, or about misconceptions or gaps in a student's knowledge. However, there are wider ways of involving AI in teaching. For example, AI-ED is sometimes defined as: *the use of AI methodologies and AI ways of thinking applied to discovering insights and methods for use in education, whether AI programs are involved at the point of delivery or not* (based on Naughton, 1986). In practice, these contrasting approaches to AI-ED form a continuum. This paper reviews work at both extremes of the definition, and at points in between. What all of the work has in common is that the design principles of the systems are substantially derived from, and expressed in, the language of Artificial Intelligence (Self, 1995).

## 1.2 Music as an open-ended domain: problem seeking vs. problem solving

One useful distinction in AI-ED is between well formalised domains ('domain' means subject area to be taught) and more open ended domains. In relatively well formalised domains such as arithmetic and Newtonian dynamics, there are clear goals, correct answers, and reasonably well understood criteria for success. In open ended domains such as music composition, there are in general no clear goals, no criteria for testing correct answers, and no comprehensive set of well-defined methods. Rittel and Weber (1984) describe problems in such domains as 'wicked problems'. In such domains, there cannot be, in general, definitive formulations of problems, let alone of answers. In wicked domains such as music composition, learners must not just solve problems, but also *seek problems* (Cook, 1994). The term 'problem seeking' is used in various disciplines, for example, in architectural design (Pena, 1987) and animal behaviour (Menzel, 1991). Cook (1994) imported the term into AI-ED particularly in the sense used by the philosopher Lipman (1991b), developed from Dewey (1916). In this sense, problem seeking refers to a reflective approach where:

- problems are treated as ill-defined and open ended,
- there is a continual intertwining of problem specification and solution,
- there are few clear criteria for completion,
- context greatly affects the interpretation of the problem,
- problems are always open re- interpretation and re-conceptualisation.

Music composition and expressive music performance in general are replete with problem seeking. There is no generally applicable goal or problem to be solved, except perhaps "compose something interesting" (Levitt, 1985) or "perform this with feeling". The learning composer must find or create goals or problems to solve, which may need to be revised, modified or rejected as she evaluates intermediate solutions. Although the term 'problem seeking' is new in AI-ED (Cook 1994), ways of characterising aspects of this kind of activity have been discussed in AI by Minsky (1981) and Levitt (1985), in Cognitive Science by Johnson-Laird (1988a, 1988b, 1991), and in AI-ED by Holland(1987, 1989) Holland & Elsom-Cook (1990), Sharples(1983), Smith (1994) and others.

## 2. Early Computer-Aided Instruction in Music Education

Music education programs that use AI not at all, or negligibly, are worth considering briefly as a background to AI-ED approaches. Historically, the first use of computers in teaching music, and most other subjects, was usually associated with the theory of learning *behaviourism*. Such systems (branching teaching programs) stepped through essentially the following algorithm. (O'Shea and Self, 1983).

- Present a 'frame' to the student i.e.
  - a) Present the student with pre-stored material (textual or audio visual).
  - b) Solicit a response from the student.
- Compare the response literally with pre-stored alternative responses.
- Give any pre-stored comment associated with the response.
- Look up the next frame to present on the basis of the response.

A representative example of this kind of system in music was the GUIDO ear-training system (Hofstetter, 1981). Branching teaching programs are relatively rigid, and cannot respond to the user in any way that has not been more or less explicitly pre-planned by the author. This tends to limit this approach to relatively simple subject matter, or at any rate, simple treatments.

A slightly more sophisticated version of branching teaching allows questions and answers to be generated from a template instead of being pre-stored. For example new sequences of chords to be identified for ear training can be generated by a chord grammar (Gross, 1984). The student's results can be used to control the difficulty and subject of the next example according to some simple pre-specified strategy.

### 2.1 Multimedia and Hypermedia approaches

The possibilities of multimedia presentation and hypermedia (Kommers, Jonassen & Mayes, 1992) have transformed music education software, giving much of it a quite different emphasis from the early behaviourist teaching programs. Recent good quality educational hypermedia music programs include Seventh Heaven, Ear Trainer, Interval and Listen, all aimed at giving practice in recognising or reproducing intervals, chords or melodies. MacGAMUT simulates classroom dictation exercises, including detailed marking of exercises. MiBAC Music Lessons, Perceive and Practica Musica offer extended ear training including scales, modes, durations, tunings, and flexible melodic dictation. Practice Room offers tuition in basic music theory. For more information on these and many other programs, see Yavelow (1982). Ear training and music theory are popular targets for non-AI music education programs, since the subject matter is relatively clear cut and non-problematic. Of course, many generally useful musical computers tools are also applicable to education (Yavelow, 1992; Roads, 1996). These include music editors, sequencers, analysis tools, innovative musician interfaces, computer - aided composition tools and multimedia reference CD-ROMs on masterworks. A good example of the later is Voyager's (1989) interactive CD-ROM for Beethoven's Ninth Symphony. We will briefly treat a few of these tools that involve AI later in the paper.

## 3. 'Classical' Intelligent Tutoring Systems for Music

Roughly speaking, the history of AI-ED can be divided into two periods, the 'classical' period, from about 1970 to about 1987, and the 'modern' period, from about 1987 to the present day. In the classical period, the most influential idea was the three component 'traditional' model of an Intelligent Tutoring System, abbreviated to ITS. This model was sometimes extended to a four component model - we will discuss each component in turn shortly. After 1987, the centre of gravity in AI-ED was widely deemed to have shifted to finding ways around the limitations of this traditional model (Self, 1995). In fairness, these limitations were actually a focus of much earlier research (Wenger, 1987), and the traditional ITS model remains influential and useful to the present day. However, this rough division helps to make sense of a widely felt shift in research emphasis.

The traditional model of an Intelligent Tutoring System (ITS) focuses on three main components, each of which, loosely speaking, can be considered a separate 'expert' system. More precisely, the traditional ITS (Sleeman and Brown, 1982) involves three modular AI components, each with its own area of expertise. The first component, the domain model, is an expert on the subject matter being taught. So, for example, in the case of a harmonisation tutor, the domain expert itself would be able to perform harmonisation tasks. This capability is considered vital if the system is to be able to answer unforeseen questions about the task in hand. The second standard component is a student model, whose purpose is to build up a model of the student’s knowledge, capabilities and attitudes. In principle, this allows the system to vary its approach appropriately for the individual student. Student modelling can be done in more or less refined ways. In the simplest case, the student model may be, in essence, a simple checklist of skills. This is sometimes modelled as an overlay, i.e. a tick-list of the elements held in the domain model. Thus, under this scheme, an individual student’s expertise is modelled as a subset of the expertise of the domain model. In more sophisticated systems, the student model might be a deliberately distorted or faulty ‘expert’ system, whose ‘errors’ are intended to mirror a student’s misconceptions. The faithful diagnosis of a student’s knowledge, skills and beliefs and their subsequent representation is, in general, a hard AI problem. One partial way around the diagnosis problem is to ask the student explicitly about their capabilities, limitations, previous experience, and so on. A more stringent approach is to set the student carefully crafted diagnostic tasks at intervals and to use the results to construct the student model.

The third component of the traditional ITS is a teaching model, which has expertise about teaching. Typically, this might consist of a set of teaching strategies ranging from styles such as ‘Socratic tutoring’, ‘coaching’, and ‘teaching by analogy’ (Elsom-Cook, 1990), to simply letting the student explore the available materials unhindered, with or without the guidance of a human teacher. Not all Intelligent Tutoring systems have all three components (the fourth component, if present, is an interactive user interface designed carefully for the tasks concerned). In practice, many ITS’s focus on just one or two components, and omit or greatly simplify the other components. In particular, most ITS’s in music have tended to focus on the expert or student model. Irrespective of the emphasis, virtually all ITS’s need to have explicit, formalisable knowledge of the task. Of course, many skills in music correspond to wicked problems and are very resistant to explicit formalisation. This narrows the possible areas of application in music for the traditional ITS model down to a limited number of areas. For example, one of the few musical topics for which relatively detailed, explicit rules of thumb can be found in textbooks is harmonisation. But even here, the traditional ITS approach may not necessarily work effectively. One of the clearest examples of the potential and limitations of the ITS approach for music can be found in two systems from the classical ITS period, Vivace and MacVoice.

### 3.1 Vivace: an expert system for harmonisation

Vivace is a rule-based expert system for the task of four-part chorale writing, created by Thomas (1985). Although not in itself a full ITS, it formed the basis for one. Vivace takes as input an eighteenth century chorale melody and writes a bass line and two inner voices that fit the melody. The system employs rules and guidelines for harmonisation, drawn from text books, abstracted from the practice of past composers. It is useful to categorise these rules loosely into four types: firm requirements, preferences, firm prohibitions and less firm prohibitions.

Three specific problems of principle can be identified for any human or machine trying to harmonise on the basis of the rules. The first problem is well known: that it is quite possible, indeed common in beginners' classes, to satisfy all of the formal rules and yet to produce a piece of music which is correct but aesthetically unsatisfactory. The second problem is that most of the guidelines are prohibitions rather than positive suggestions. Milton Babbit, discussing counterpoint, observes that "... the rules ... are not intended to tell you *what* to do, but what *not* to do" (Pierce, 1983). To put it another way, if harmonisation is viewed as a typical AI 'generate and test' problem, the rules constitute weak help in the testing phase, but little help in well-focused generation. The third problem is that it is often impossible to satisfy all of the preferences at once - usually some preference rules have to be broken; but traditional descriptions of the rules do not assign the preference rules a clear order of importance. In fact it is not at all clear that any fixed priority orderings would make sense.

Despite these problems, it is possible to write a rule-based system that implements the text book rules. In principle, a traditional ITS could use these rules to criticise students' work, and to serve as a model of the expertise they are supposed to acquire. But how relevant or useful would such a tutor be, given the limits outlined above? Thomas used the tutor to *illuminate the limitations* of the theory, as outlined below.

By building and experimenting with Vivace, Thomas was able to establish clearly that typical text book rules are an inadequate characterisation of expert performance of the task. For example, Thomas discovered that if tenor and alto parts are written using only conventional rules about range and movement, the tenor voice would often absurdly move to the top of its range and stay there. Thomas posited that there must be a set of missing rules and meta-rules to fill the gaps, and set about trying to find them, using Vivace as an experimental tool. In each experiment, Thomas had to decide, on the basis of intuition, whether a result was musically acceptable or not. Thomas was able to make explicit many new detailed considerations about harmonisation that were previously only tacit, and found out that many of the traditional rules were overstated or needed redefining. As well as uncovering new guidelines about matters of detail, Thomas was able to make explicit knowledge about the task at a more strategic level. Thomas and her human pupils formulated a number of heuristics for 'what to do' as opposed to 'what not to do'. Experience with Vivace also underlined for Thomas the need to make human pupils aware of high level phrase structure, before diving into detail of chord writing. Having experimentally discovered new explicit knowledge about the task, as a result of 'teaching' her expert system, Thomas used this knowledge as a basis for writing a new teaching text for the task. Part of this knowledge was also used in a simple commercial ITS, MacVoice, which criticises students' voice-leading.

### 3.2 MacVoice: an intelligent critic for voice-leading

MacVoice is a Macintosh program based on the expert system Vivace. It criticises voice-leading aspects of four-part harmonisation. The MacVoice interface includes a simple music editor. It is possible to input notes in any order, for example a chord at a time, or a voice at a time, or notes in any fragmentary fashion. As soon as any note is placed on the stave, MacVoice displays its guess as to the function of the corresponding chord in the form of an annotated Roman numeral (figure 1). Two important limitations of MacVoice are as follows: firstly, all notes must be of the same duration (so that the chords form homophonic blocks); and secondly, the piece must be in a single key. Apart from saving and loading files or erasing the stave, there is only one other menu function, called 'voice-leading'. When this is selected, a rule base of voice-leading rules inspects the harmonisation, and then provides a list of voice-leading errors. MacVoice is relatively flexible in use, since it can be used on exercises where any combinations of voices or notes are already filled in.

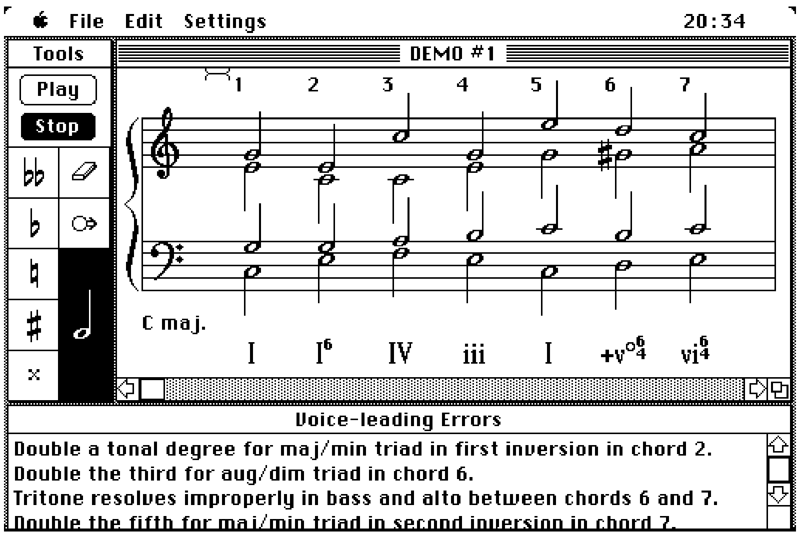


Figure 1. MacVoice.

MacVoice was the first program of its kind, and has been used practically at Carnegie Mellon University. MacVoice 2.0 only points out errors, it does not give positive strategic advice. Neither does MacVoice address the sensibleness or otherwise of the chord sequences involved. Topics for further research might be to show *visually* what the voice-leading constraints are, or what the preferred possibilities are at any point. Perhaps as the task became more constrained, candidate notes might be shown (on request) in different shades of grey corresponding to more or less likely possibilities. Another possibility for further research would be to try to abstract or group the rules according to a smaller number of abstract principles that they appear to serve. Such principles could be used to inform and illuminate the tutor's criticisms. The problem of trying to separate lower level knowledge from higher level knowledge in expert systems, and to provide graphic windows on a tutor's inference processes are both important problems in traditional ITS's. In the next section we look at another ITS from the 'traditional' period, Lasso, that offers several useful contrasts with Vivace/MacVoice.

### 3.3 Lasso: an Intelligent Tutoring System for 16th century counterpoint

Lasso is an intelligent tutoring system for 16th century counterpoint, as formalised by Fux (1725), limited to two voices. Rather than equivocating about whether the goal of the system is to encourage good composition, or to encourage scholarly fidelity to a historical style, Newcomb, with commendable honesty, notes that his rules are intended to serve as simple and consistent guidelines to help students know what is required to pass exams. Like Thomas, Newcomb found that the process of codification of the necessary knowledge required going beyond rules and guidelines given in text books. Unlike Thomas, Newcomb went about this in a probabilistic manner, analysing scores to find out such 'facts' as "the allowable ratio of skip to non-skip melodic intervals" and "how many eighth note passages can be expected to be found in a piece of a given length" (Newcomb, 1985).

Lasso cannot write counterpoint itself (Newcomb, 1985), and the knowledge used for criticising the students work is not coded particularly explicitly, being encoded as branching procedural code. Unvarying canned error messages, help messages, and congratulatory messages are used. Where the rules go beyond traditional rules, they are based on a way of characterising style that relies on counting and legislating about the frequency of particular low level features. Although this limits its extensibility, Lasso is in many ways impressive. It has a high quality music editor associated with it; tackles a complex musical paradigm; and has been used in real teaching contexts. However, there are some inherent problems. Firstly, the rules are at a very low level, and there are a lot of them. This is reflected by the fact that there needs to be a system rule preventing more than one hundred and twenty-seven comments being made about any given attempt to complete an exercise! To give a flavour of the rules, typical remarks by Lasso include;

> "A melodic interval of a third is followed by stepwise motion in the same direction."
> "Accented quarter passing note? The dissonant quarter note is not preceded by a descending step."
> (Newcomb, 1985)

A student could easily be continually overwhelmed by the quantity of relevant help text required to put in context a myriad of low-level criticisms. Students complained that "it was so difficult to satisfy LASSO's demands that they were forced to revise the same exercises repeatedly" (Newcomb, 1985). As in the case of MacVoice, one way of tackling this problem might be try to code explicit general principles governing the low-level rules. Such codified principles might be used to cut down the number of low-level comments in any particular case and replace them by a smaller number of relevant but more general observations. Scope for other future research in this kind of system, while still making use of traditional ITS techniques, might include the provision of explicit teaching rules and an explicit user model. Such developments might help the tutor to reason about when not to say things; to decide to concentrate on one fault at a time in a principled way; or to reason explicitly about when and how to offer strategic advice. Related systems for teaching music include Schaffer’s (1991) Harmony Coach; Sorisio’s (1987) intelligent tutor the MUSES for music theory; Sanchez, Joseph, Dannenberg, Miller, Capell & Joseph’s (1987) intelligent tutor ‘Piano Tutor’ developed at Carnegie Mellon University; Fenton’s (1989) Intelligent Tutoring System for Music, and Camurri et al’s (1991) System for Intelligent Composer's Assistance.

### 3.4 Concluding remarks on the classical approach to Intelligent Tutoring Systems for music

To a greater or lesser extent, the traditional ITS approach assumes an *objectivist* approach to knowledge (Self 1995). That is to say, such systems are generally based on the assumption that there exists a well-defined body of relevant knowledge to be taught, and that it can be carved up into more or less precise concepts and relationships. This works, to some extent, with 16th century counterpoint and four-part harmonisation, as traditionally taught. In a more open ended context, an objectivist approach is of more limited value. Even in artificially limited domains, the explicit teaching of rules drawn from existing practice is not necessarily a good approach. Verbal definitions of a musical concept are often very impoverished compared with the rich multiple meanings they must come to have for an experienced musician. It is all very well to define a dominant seventh, for example, to a novice in terms of its interval pattern and then give some rules for its use, but to an experienced musician, the 'meaning' of a dominant seventh varies greatly depending on the context. Getting the novice to obey any set of rules is really far less important than making them aware of, and able to manipulate intelligently, the structures and expectations that are available to the more experienced musician. What is needed is not so much explanations of rules, as structured sequences of experiences that make the novice more aware of musical structures, more able to manipulate them intelligently, and more capable of formulating sensible musical goals to pursue.

## 4. The Logo Philosophy: Open-ended Microworlds

A contrasted idea from the classical period of AI-ED, which is just as influential as the notion of an ITS is the Logo approach (Papert, 1980). The Logo philosophy has particular attractions in open-ended areas such as music. The Logo approach centres on the notion of an educational microworld. An educational microworld is a well-structured, open-ended environment for learning that focuses on some problem domain, but without, in general, specific lessons built in. Microworlds associated with the Logo approach need not involve much, or indeed any AI at the point of delivery, but their design tends to be strongly influenced by AI methodologies and tools. Often, but not always, the microworld is built using a simple version of an AI programming language, and students are encouraged to write or modify programs written in this language as a means of exploring the domain. Indeed, Logo doubles as the name of a simplified AI programming language based on Lisp, used for just this purpose. As Green points out (Holland, 1989) it is important to distinguish at least three distinct elements in the Logo approach: Logo (and similar languages) as a programming tool; Logo as a vehicle for expressing various AI theories for educational purposes; and Logo as an educational philosophy, as we will now briefly explore.

Early work on Logo was focused on mathematics learning, poetry and music. In one of the early versions of Logo, children were encouraged to rearrange or modify melodic phrases to produce new melodies. The associated learning philosophy aimed to build up children's' understanding by getting them to envision or pre-hear an expected result, work out how to achieve it, and then 'debug' their emerging abilities when unexpected results were obtained. This learning philosophy drew on diverse sources, including the psychologist Piaget's notions of how children construct their own knowledge through play.

The microworlds involved in a Logo approach can be more or less complex. In some cases, students are provided with a simplified version of an AI model of some problem domain under study. For example, in the case of music composition, generative grammars can be used as models of particular composition techniques, and used to generate fragments of illustrative materials. Students may use the supplied programs to explore, criticise and refine their own (or someone else's) model of some process. Interestingly, this way of using an educational microworld resembles the research methodology for developing AI theories using AI microworlds (Desain and Honing, 1992):

Notice that none of the three components of the traditional ITS model need be present using the Logo approach. In practice, students tend to need careful guidance from teachers to make headway with Logo systems - otherwise there is a danger that they will become stuck in some small area of technique without appreciating the wider possibilities. The educational philosophy associated with Logo has been applied to music in detail at various different levels and in various different ways, as explored below.

### 4.1 Bamberger's Music Logo System

Jeanne Bamberger's Music Logo System (1986, 1991) is a version of Logo adapted to work with a sound card or synthesiser. Music Logo uses programming elements called functions to structure and control musical sounds. Programs are sequences of function calls. The central data structures in Music Logo are lists of integers representing sequences of pitches and durations, which may be stored separately. Pitch lists and duration lists can be manipulated separately before being played by a synthesiser. So for example, to play E above middle C for 40 beats, then middle C for 20 beats, then G for 20 beats, an expression like the following might be used.

```
play [e c g] [40 20 20]
```

Simple programming constructs such as **repeat** can easily be seen by beginners do useful musical work. More generally, note lists and duration lists may be manipulated separately using arithmetic and list manipulation functions. Complex musical structures may be built up using such programming features as recursion and random number generators. List manipulation functions that correspond to common musical operations are provided. For example, one function takes a duration and pitch list and generates a specified number of repetitions of the phrase shifted at each repetition by some constant specified pitch increment - creating a simple sequence (in the musical sense of the term). Other musical functions and their effects include **retrograde** which reverses a pitch list, **invert**, which processes a pitch list to the complementary values within an octave, and **fill**, which makes a list of all intermediate pitches between two specified pitches.

Bamberger suggests many simple exercises that are variations on the basic activity of iteratively manipulating the list representations to try to reproduce some previously imagined musical result, or conversely making formal manipulations on lists and procedures and trying to guess the musical outcome. In many ways these techniques are similar to reflective educational techniques suggested by Laurillard (1993) for general use in higher education. Bamberger stresses the importance of the 'shocks' and learning experiences precipitated by two particular classes of phenomena. Firstly, small manipulations of the duration list often produce radically changed perceptions of where phrase boundaries occur in melodic and rhythmic fragments. Secondly, there is an unpredictable disparity between degree of change in the data structure and the degree of perceived change produced. In principle, Bamburger's Music Logo could allow students to focus on the explicit manipulation of any kind of musical structuring techniques. In practice, this work has tended to focus on simple, small scale musical structures such as motives, and their transformation.

### 4.2 The LOCO series of microworlds

A particularly elegant set of systems applying the Logo philosophy and techniques to music is a series of microworlds and tools developed by Peter Desain and Henkjan Honing. The series starts with LOCO (Desain & Honing, 1986, 1992), followed by POCO (Honing, 1990), Expresso (Honing, 1992) and LOCO-Sonnet (Desain & Honing,1996). All of these microworlds reflect careful thought about how AI methodologies can be applied to music education. LOCO, like Bamberger's Music Logo, is a set of extensions to Logo for dealing with music composition. The central component is a general purpose set of tools for representing sequences of musical events, flexible enough to be interfaced with any output devices or instruments. The system is also flexible enough to take input from more or less any composition system. A set of microworlds is provided, each of which offers tools for a series of generally useful, style-independent composition techniques - in particular stochastic processes and context free music grammars. The kinds of musical object provided are essentially just two: notes and rests. LOCO has an elegant, simple and general time structuring mechanism. Two relations are provided, PARALLEL and SEQUENTIAL which can be used to combine arbitrary musical objects. SEQUENTIAL is a function that causes musical objects in its argument list to be played one after another. PARALLEL is a function that causes its arguments to be played simultaneously. It is straightforward to nest any number of PARALLEL structures within a SEQUENTIAL structure and vice versa. Before any data object is played, SEQUENTIAL and PARALLEL objects are treated as items of data that can themselves be computed and manipulated. The result of this framework is that arbitrary time structuring can be applied with a high degree of flexibility. As already noted, LOCO provides primitives for composing using stochastic processes and context free grammars. Depending on how variables are defined, their subsequent evaluation may produce various effects, including;

- a random choice among its possible values,
- a choice weighted by a probability distribution,
- a random choice in which previous values cannot recur until all other values have been picked,
- selection of a value in a fixed circular order.

The above components are easily put together via composition (in the mathematical sense) of functions (figure 2). For example, the value of an increment could be specified as a stochastic variable, producing a variable that performs a Brownian random walk. Brownian variables could then be used, for example, as arguments in commands to instruments within some time-structured framework. Techniques such as these can be used to construct concise, easy-to-read programs for transition nets, Markov chains and other stochastic processes. In each case, the precise operation of the program can be modified using the full power of a general purpose programming language. See Ames (1989) for a wide-ranging discussion of the compositional uses of Markov chains. In a similar way, rewrite rules for context-free grammars can be implemented almost directly, for example to produce rhythms. Competing rewrite rules can be assigned probabilities giving rise to so-called 'programmed grammars'. The system gives elegant support for both discrete and continuous aspects of music, as discussed in Honing (1990).

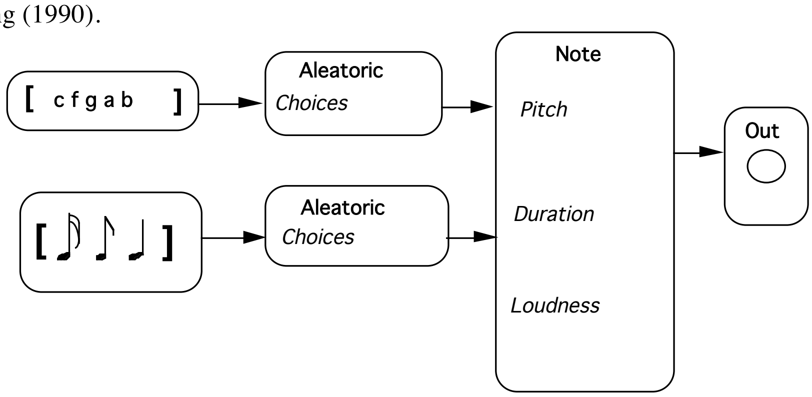


Figure 2. A random choice of pitch and a random choice of duration combined in LOCO-Sonnet to produce a stream of notes. (Re-drawn from Desain and Honing (1996).

The primary design goals of LOCO include ease of use by non-programmers, modularity at all levels (to allow anything to be varied with minimal damage to anything else) and user-extensibility. A more recent variant of LOCO, LOCO-Sonnet, is designed to mirror LOCO but with a graphical front end. The front end draws on Jameson's (1992) Sonnet (a domain independent data flow language originally designed for adding sound to user interfaces). Sonnet has similarities with Levitt's (1986) Hookup, and to a lesser extent, Puckette's (1988) MAX.

LOCO and its variants are useful for novice and experienced programmer alike. It is a well-designed, relatively easy-to-use programming language, with only a small number of concepts that novices need to learn. Structuring and instrument interfacing conventions are modular, flexible and general. LOCO has been used successfully in workshops for novices and professional composers, and now has courseware available.

### 4.3 Concluding remarks on the Logo approach

The Logo approach is commonly associated with a view of knowledge known as constructivism. Constructivism, as a view of knowledge and of learning, asserts that even in cases where 'objectively true knowledge' exists, simply presenting it to a learner may have a limited effect on their learning. Constructivism asserts that learning arises substantially from learners' active encounters with the world, which force them to construct their own knowledge for themselves. The resulting 'knowledge' may be different from someone else's 'knowledge'. Constructivism apparently fits well with open ended domains like music since, for example, each composer ultimately appears to construct her own 'knowledge' about how to compose.

A key limitation (or perhaps strength) of the Logo approach is that it demands intensive support from a human teacher, and to some extent, the exchange of ideas with peers in order to be effective. Intelligent Tutoring Systems and the Logo approach were two of the most influential early ideas in AI-ED. As the limitations of both the Logo approach and the ITS approach became more widely noted, systems that combined characteristics of both systems became a focus of research. Such systems are often called Interactive Learning Environments (abbreviated to ILE). We will look at several Interactive Learning Environments (ILEs), after a quick look at AI-based tools.

## 5. AI-based tools with Applications in Education

There are numerous musical tools and kits employing AI whose purpose is not primarily educational, but which nevertheless have clear educational applications. It is useful to briefly consider some of these systems. They form a large group with many overlaps, so we will limit ourselves to a few representative examples. There are a number of musical programming languages, environments and tool kits based on general purpose AI languages (often LISP or CLOS), that have a technical content closely related to that of the Music Logo systems described above. The associated philosophy of use, however, may be very different. One example is the commercial system Symbolic Composer (for the Macintosh and Atari). This is built on Common Lisp, and includes a vast library of functions, including neural net facilities, designed for processing, transforming and generating musical data and processes. This system, like many others in this section, is aimed principally at composers and researchers. Common Music (Taube, 1991) includes a pattern-oriented composition language, while CLM (Common Lisp Music) for the NeXT is an environment is specialised for synthesis, signal processing, and aspects of composition. Descriptions of these and similar systems can be found in Roads (1996).

The Smalltalk culture, which has many links with AI culture also offers many environments and tool-kits full of educational potential. Pachet's (1994) MusES environment, implemented in Smalltalk 80, which is aimed at experimenting with knowledge representation techniques in tonal music is a good example of such a system. MusES includes systems for harmonisation, analysis and improvisation. Finally, it is useful to consider an example of a particularly successful commercial program. Band in a Box (Binary Designs, 1996) takes as input a chord sequence, and as output can play an accompaniment based on the chord sequence in a wide variety of styles. At one time this would have required AI techniques (Levitt, 1985), but in fact the system uses conventional methods.

## 6. Supporting Learning with Computational Models of Creativity

### 6.1 A cognitive support framework based on a constraint-based model of creativity

> "I noticed that the [drawing] teacher didn't tell people much.... Instead, he tried to inspire us to experiment with new approaches. I thought of how we teach physics: we have so many techniques - so many mathematical methods - that we never stop telling the students how to do things. On the other hand, the drawing teacher is afraid to teach you anything. If your lines are very heavy, the teacher can't say "your

lines are too heavy" because *some* artist has figured out a way of making great pictures using heavy lines. The teacher doesn't want to push you in some particular direction. So the drawing teacher has this problem of communicating how to draw by osmosis and not by instruction, while the physics teacher has the problem of always teaching techniques, rather than the spirit of how to go about solving physical problems" Feynman (1986)

"John and I ... were quite happy to nick things off people, because ... you start off with the nicked piece and it gets into the song ... and when you've put it all together ... of course it does make something original" Paul McCartney quoted in (Moore, 1992)

The two AI-ED approaches from the classical period of AI described in the previous section (ITS and Logo) are still very influential, but both approaches have their limitations. In particular, traditional ITS's do not work very well in problem seeking domains, while Logo type approaches require a lot of guidance from a human teacher to be effective. MC (Holland, 1989, 1991; Holland & Elsom -Cook, 1990) is a general framework for interactive learning environments in open-ended domains, which investigates one way of addressing these problems. 'MC' is an acronym both for 'Meta Constraints' and 'Master of Ceremonies'. The framework supports a variety of guidance strategies (Holland, 1989; Holland & Elsom-Cook, 1990), but we will focus here on the domain model rather than the teaching component. The current version of the system is aimed at teaching *ab initio* students to compose sensible and interesting tonal chord sequences, with particular reference to popular music and jazz harmony. The system employs a very general cognitive theory of Harmony (Balzano, 1980) and is applicable in principle to any kind of tonal music, and some non-tonal music.

Johnson-Laird's view of creativity is related to ideas expressed by Levitt (1981, 1985), Sloboda (1985) and Minsky (1981). For our purposes, we need only consider two elements from Johnson-Laird's definition. The first element is the assumption that creative tasks cannot proceed from nothing: that some initial building blocks are required. The second element is the assumption that a hall-mark of a creative task is that there is no precise goal, but only some pre-existing constraints or criteria that must be met (Johnson-Laird, 1988a). From this starting point, the act of creation can be characterised in terms of the iterative posing and eventual satisfaction of constraints by the artist. The artist may at any time add *new* constraints to the weak starting criteria. At each iteration, results are tested against (possibly tacit) acceptance criteria. Sometimes, and in some domains, it may be acceptable to sacrifice a pre-existing constraint or criterion in order to meet new constraints imposed by the artist: Sloboda (1985) puts this very clearly;

".. we will find composers breaking .. rules [specifying the permissible compositional options] from time to time when they consider some other organisational principle to take precedence." (Sloboda, 1985)

### 6.1.1 The nature of constraints in music

Constraints in music seem to fall broadly into three types: some based on fostering perceptual and cognitive conditions for effective communication, others based on cultural consensus and yet others introduced from scratch by the artist. Examples of the first kind of constraint appear in research on western tonal music, such as Balzano (1980), Minsky (1981) and McAdams and Bregman (1985). These focus respectively on harmony, metre and, amongst other things, voice leading. Each of these pieces of research emphasises how various widespread features of music appear to have an important role in fostering perceptual and cognitive conditions for effective communication of structure.

The second class of constraints is of a cultural, historical nature. The nature of music is not only affected by the structure of musical materials and how these interact with our perceptual and cognitive faculties. Also important is listeners' familiarity with the way in which composers happen to have used these materials previously. When listeners hear a new piece of music, cognitive theories of listening posit that the music must be chunked in various ways to cope with memory and processing limitations (Sloboda, 1985). The kind of chunking that can be done by a particular listener depends not only on the abstract nature of materials such as harmony, but on the practices, genres and particular pieces of music with which the listener is already familiar. Levitt puts the connection between stylistic constraints and those introduced by an individual composer neatly (Levitt, 1981);

"Effective communication requires musicians to repeat structures frequently within a piece and collectively over many pieces. Usually we view 'musical style' and 'theme and variation' as utterly different. Computationally and socially they are similar things with different time spans; style tries to exploit long term 'cultural memory', while theme and variation exploits (sic) recent events. In either case, the considerate composer uses an idea of what is already in the audiences head to make the piece understandable." (Levitt, 1981)

### 6.1.2 A minimal computational model of creativity

Putting these insights together, we posit that open ended creative activities can be modelled by the following process. Given a set of building blocks,

Choose a goal.
Select constraints.
Iterate the following process:

- apply the constraints to generate a result,
- test the result against (possibly tacit) acceptance criteria,
- adjust the constraints (or possibly the criteria or goal)
  until acceptance criteria are sufficiently closely met.

### 6.1.3 Components of the MC framework

The MC framework provides a set of modular but interacting components (figure 3) that act as a cognitive support tool for creative processes. As applied to the domain of composing tonal chord sequences, the key components of MC (Holland, 1989, 1991; Holland & Elsom-Cook, 1990) are as follows.

- A constraint-based planner (PLANC).
- Constraint based representations of basic musical materials such as modes, scales, chords, etc. for use as raw materials by the planner.
- A family of harmonic plans or prototypes, for example 'return home', 'hook', 'modal harmonic ostinato', 'moving goalpost'(Holland, 1989), that can be used with the planner and raw materials to generate prototypical harmonic sequences. Each plan has a number of controlling variables. Choosing values for a variable selects musical materials, techniques or strategies for the plan. Each instantiation produces, in general, a different chord sequence (or often many different sequences, each with a family resemblance).
- An extensible corpus of existing pieces. Each piece is linked to one, or often more plans that can generate its chord sequence, and to relevant styles.
- Constraint-based descriptions of musical styles expressed as common features in the constraint descriptions of two or more songs.
- A highly interactive direct manipulation microworld (described in the next section) based on a cognitive theory of harmony. This allows users to concretely manipulate all of the basic elements of harmony used in the pieces; intervals, chords, voicings, chord sequences, modulations, etc., in a form highly accessible to beginners (Holland 1989, 1994).

The plans make use of spatial primitives derived from Balzano's theory (1980) of tonal harmony. The interactive microworld associated with MC allows beginners to manipulate and become intuitively familiar with harmonic materials expressed in this form. The Harmonic plans correspond loosely to harmonic patterns in popular music such as those noted by Cork (1988) and Moore (1992), and are related to those in Pratt (1984). As Moore notes, far from merely consisting of a few stock formulae, harmony in this domain is sufficiently complex that no generative grammar for the area yet exists, though grammars for sub-genres do exist (Steedman, 1984). PLANC uses a functional harmony notation, though the standard rules of classical functional harmony are not fully applicable. Extensive reference is made to modal harmony. Indeed, one source of power in the plans is generalisation over tonal and modal harmonic sequences moving towards a tonal centre.

A typical harmonic plan is the 'return home' plan. A return home involves establishing a tonic (in a way that depends on whether the context is tonal or modal; moving to another root; and then moving back home in diatonic fifths (in the tonal case) or scalewise (in the modal case) in a direction that may depend on the choice of mode. The home chord may or may not need to be explicitly stated at the beginning of the chord sequence, depending on what other musical resources are available to perceptually communicate its presence. If available, these resources may be represented explicitly in the plan. If the underlying alphabet of chords in force at a given time is restricted, then this further constrains the chord trajectory. To facilitate the generalisation over tonal and modal systems, Mehegan's (1959) notation is used. Thus chord quality is indicated not by an upper case/lower case distinction, or by annotation, but is assumed to conform to the degree of the root, and scale or mode in force, unless explicitly annotated otherwise.

One simple but widely applicable plan, not involving any modulations, is the 'return home' plan. Examples of return home chord sequences include the following.

| | | |
|---|---|---|
| "I got rhythm ' (Gershwin) | (Major) I VI II V I | (in scaletone sevenths) |
| "Abracadabra" (Steve Miller) | (Minor) I IV V I (restricted alphabet of roots in force) | |
| "The Lady is a tramp" (Rogers and Hart) | (Major) I IV VII III VI II V I | (in scaletone sevenths) |
| "Street Life" (Randy Crawford) | (Minor) I IV VII III VI II V dom7 I | (in scaletone sevenths) |
| "Easy Lover" (Phil Collins) | (Aeolian) VI VII I | (in scaletone triads) |
| "Isn't she lovely" (Stevie Wonder) | (Major) VI II V I | (in scaletone ninths) |
| "Out-Bloody-Rageous" (Ratledge) | (Dorian) III II I | (in scaletone sevenths) |

(See paragraph above for details of the chord notation used.)

It is important to note that this characterisation of these chord sequences represents only one of several viewpoints afforded by the system. Most of the sequences can also be characterised in other ways by other plans. For example, the last three chord sequences in the list above are also characterised by the 'modal harmonic ostinato' plan, and the first sequence can be generated by the 'hook' plan. In fact, 'interesting' chord sequences tend to be those that can be characterised by several plans simultaneously. Each viewpoint on each song gives a different tree of 'nearest neighbours' in the corpus, and emphasises different structurally important features. This multiplicity of viewpoints, illustrated by the corpus, and coupled with the ability to iteratively modify and generate new pieces in a principled way is an important source of power in the system. Plans are expressed using a constraint representation similar to that of Levitt (1985). See Holland (1991) for detailed analyses of the plans mentioned.

The basis of computation of the musical planner PLANC can be characterised formally as follows. (Non-mathematicians may wish to omit the formal definition and simply take the term 'constraint satisfaction' in its everyday sense.) The satisfaction of a musical plan is a **special case of the formal class of constraint satisfaction problems** (CSP). Van Hentenryck and Dincbas (1987) give a formal definition of this class of problems as follows.

> Assume the existence of a finite set I of variables $X_1, X_2, \ldots, X_n$ which take respectively their values from the finite domains $D_1, D_2, \ldots, D_n$ and a set of constraints. A constraint $c(X_{i_1}, \ldots, X_{i_k})$ between k variables from I is a subset of the Cartesian product $D_{i_1} \times \ldots \times D_{i_k}$ which specifies which values of the variables are compatible with each other. A solution to a CSP is an assignment of values to all variables which satisfy all the constraints and the task is to find one or all the solutions.

Van Hentenryck and Dincbas (1987) go on to give a clear statement of the simplest (though not the most efficient) way of solving constraint satisfaction problems using logic programming languages. They point out that "Given a particular CSP, it is sufficient to associate a logic program with each kind of constraints (sic) and to provide a generator of values for the variables."

PLANC will provide default values for any variable of any plan via such generators. The generators allow beginners to experiment with the elements of a plan in isolation, or in any combination. This makes it possible to explore particular effects, without being pressed to specify elements that they may not yet understand or wish to focus on. Often very sparse specifications are used (e.g. just specifying the value of a single variable in a plan). The interaction of the constraints and the default value generators is such that the defaults selected may vary widely (and intelligently) between specifications. The planner permits the student to work bottom up or top down. That is to say, in generating new sequences it is possible to specify low level matters of detail while postponing decisions about high level strategic choices as well as vice versa. This reflects the varied ways in which composers seem to like to work (Sloboda 1985). MC provides a suitable infrastructure for a wide range of teaching strategies appropriate to problem seeking domains (Holland 1989). However, to understand the basics of how the components of MC interact, it is useful to focus on one of the simplest, open-ended, user-directed learning strategies that MC supports.

### 6.1.4 Analysis by recomposition

One way of using the system with beginners is to focus on an existing piece of interest and let the student 'recompose' it in a variety of ways. To see how this can be done, recall that each piece is linked to one or more harmonic plans and one or more sparse specifications for each plan that generates the piece. In effect, each generating plan, and each specification, provides an analysis or viewpoint on the piece. A user may 'recompose' a piece by selecting a viewpoint and then making minimal changes to the specification and re-generating the piece. The new result may correspond to an existing piece in the corpus (which the system could check for and remark upon), or it may correspond to a new piece. In either case the re-composition constitutes a step in exploring the nearby neighbours of the plan tree by changing the low-level materials or mid-level strategies used. Depending on the viewpoint picked, the piece will be seen to have different nearby neighbours, i.e. different pieces which are similar to the starting point as judged from that viewpoint.

The object of the exercise is neither to harvest, for their own sake, the new pieces generated, nor to learn the harmonic plans. More important is that a musically rich context in which the interaction of musical materials and existing pieces can be explored is provided by the interplay of the following elements: the corpus, the multiple viewpoints, the styles, and the new pieces generated. Depending on the preferred focus of the student at different time, the plan trees can be navigated to gain a better understanding of particular songs, particular viewpoints, or particular musical materials. Style trees can also be explored and used to vary pieces. The links between the planner, the plans, the annotated corpus, the styles and the interactive front end allow composition and analysis of existing pieces to be interleaved flexibly.

The design of MC was inspired in many respects by the architecture of the machine learning program Eurisko (Lenat and Brown, 1984). A key principle behind MC is that the meaning of musical materials emerges not in a vacuum, but from the web of different ways in which they are used in different existing pieces, as seen from multiple viewpoints. Carefully comparing similarities and differences between real pieces from different viewpoints is one perhaps of the most fundamentally useful forms of music analysis.

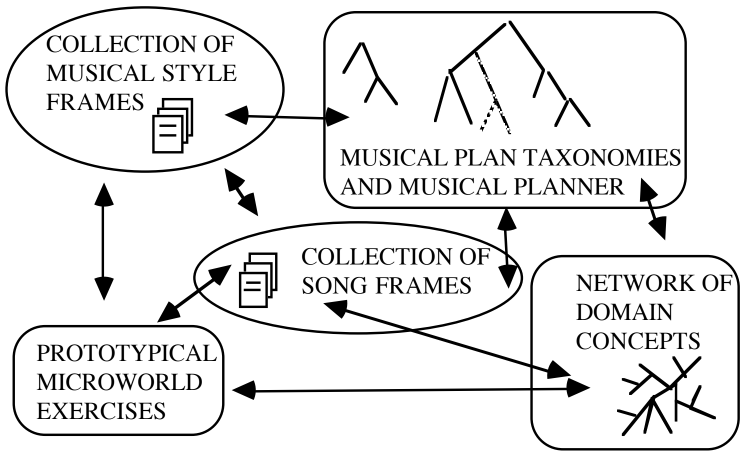


Figure 3. Kinds of domain knowledge in MC.

Three layers of knowledge in the planner interact to produce musically 'intelligent' or knowledgeable behaviour: the net of constraints, the generators and small procedural code fragments. Knowledge is recorded mostly declaratively, and independently of any particular use. Declarative programming is ‘economical with knowledge’, with the effect that it allows repeated use to be made in different contexts of simple, ordinary pieces of musical knowledge. The musical plans as implemented demonstrate how little musical knowledge is required for surprising competence in the domain, or to put it another way, how efficiently they use the limited musical knowledge that they do have. For example, one heuristic can be paraphrased as follows. "Where a modal trajectory is required to move towards a harmonic centre, but the starting side (i.e. above or below the harmonic centre) is not specified, other things being equal, prefer a starting point that ensures the trajectory will avoid the scaletone diminished chord”. Like any heuristic, it is far from universally applicable, but having been recorded just once, the system puts it to use in many contexts. For example, it helps make sense of the tendency that Dorian harmonic ostinati of a variety of lengths typically move upwards from the tonic, (e.g. Dorian I II III II I). Similarly, the Aeolian equivalents typically move downwards (e.g. Aeolian I VII V VII). The same heuristic can be usefully employed in any mode, in a variety of plans, in different contexts. More generally, PLANC shows how three layers of very simple knowledge: constraints; orderings for default generators; and small chunks of procedural knowledge; are adequate to yield competent but flexible musical behaviour. The declarative form of the constraints also makes the knowledge potentially very easy to manipulate, reason about and chunk from a variety of alternative viewpoints. MC in general is designed to allow novices to begin tackling interesting, motivating, 'high level' musical tasks as early as possible.

### 6.1.5 Limitations of current implementation of MC

The planner, several plans, numerous pieces, some styles, and a microworld for harmony have been implemented. The student model, a semantic net to allow construction of explanations, and explicit teaching modules have not yet been implemented. The current version of MC is a prototype that can be used to demonstrate all of the key principles involved, but is not a practical system. However, the framework is compatible with any of the standard teaching and modelling techniques identified in Spensley and Elsom-Cook (1988), and in principle with the more sophisticated techniques developed by Baker (1994) and Cook (1994) described later in this paper. The current implementation of the MC focuses on harmony, with some attention to metre and bass lines. However the architecture is essentially domain neutral, and is equally applicable to other dimensions of music, separately or together. Levitt (1985) constitutes one suitable basis for extending the work into melody, and Watson (1990) and Kane (1991) have carried out preliminary investigations to extend the work into rhythm. The work in the next section uses a related framework, but focuses on melody.

### 6.2 A principled constraint-based learning tool for exploring melody

Smith's (Smith and Holland, 1994) constraint-based learning tool MOTIVE for exploring melody works within a loosely similar constraint-based methodology as MC. The tool is focused on Narmour's (1989) cognitive theory of melody. The aim of MOTIVE is to support *ab initio* beginners to explore the composition of melody. The work achieves potentially very general applicability to melody, irrespective of genre, by virtue of being based on the most fundamental psychologically grounded theory of melody currently available (Narmour, 1989). Narmour's theory has known problems and limitations (Cumming, 1992), but has little competition as a theory of melody framed substantially in psychological terms. Narmour's (1989) analytical theory of tonal melody uses simple extensions to low level gestault processing theory for melodic notes to predict how a listener will break a melody up into groups of contiguous notes, and which notes will be perceived as more important than others (other things being equal). This gives rise to hierarchical parse trees which recursively reduce the melody to simpler versions, roughly analogous to Lerdahl and Jackendoff (1983) TSR trees. A central contribution of Smith's work is that, in order to be able to make use of Narmour's theory computationally, an explicit, consistent computational model of the theory had to be refined and implemented. Having done this, Smith was able to test Narmour's published hand-produced analyses for consistency against the computational version. The tests showed the theory to be internally coherent, with some gaps to be filled in, but with no fatal internal flaws. This computational model then became the central component of Smith's teaching system, MOTIVE.

MOTIVE uses a constraint based planner, similar in form to PLANC to parse melodies. Like PLANC, MOTIVE is able to 'replan' or recompose melodies by navigating trees of related melodies, while holding constant, or varying structural features of the melody at any level. Thus the 'analysis by recomposition' strategy introduced in MC can be applied, as well as other teaching strategies.

Since the empirical status of Narmour's theory is unclear, and there is no real consensus whether an adequate theory of melody has yet been found, it is not yet clear the extent to which a teaching system based on Narmour's theory will make a good practical basis for supporting beginners learning to compose melodies. This must await empirical testing. Irrespective of this outcome, it is highly likely that Smith's work will serve as a useful computational tool to help explore, modify, or refine Narmour's theory.

In the next section we will look at an interface that is based on AI theories of music perception, but where the theories are used to determine the behaviour of a highly interactive direct manipulation interface, rather than to direct a parser or a planner.

## 7. Highly Interactive Interfaces based on AI Theories

### 7.1 A highly interactive tool based on an AI theory of harmony

Harmony Space (Holland 1989, 1992, 1994; Holland & Elsom-Cook, 1990) is a highly interactive direct manipulation tool (figure 4) for learning about tonal harmony. The design of the tool employs Longuet-Higgins' (1962) and Steedman's (1972) artificial intelligence theory of tonal harmony, and Balzano's (1980) competing cognitive theory. Different versions of the tool use different version of the theory. Longuet-Higgins' (1962) theory showed how a wide range of harmonic phenomena can be characterised very concisely and clearly by re-formulating the tonal pitch system and harmonic relationships in terms of a three dimensional co-ordinate system. Balzano's competing theory posits a different, but related three dimensional co-ordinate system for pitch, based on a completely different characterisation of pitch (group theory as opposed to frequency ratios).

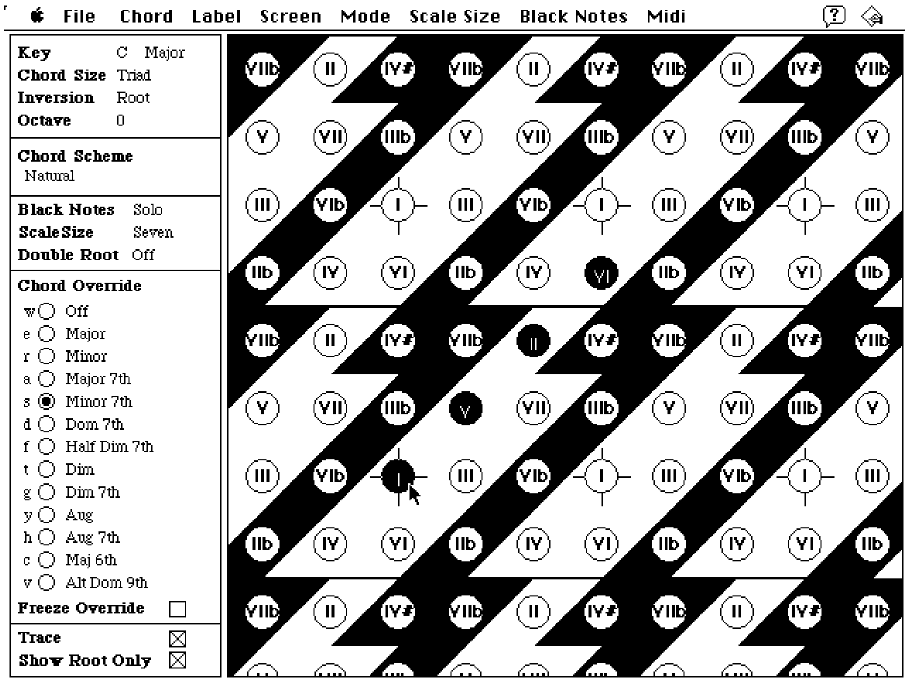


Figure 4. A Harmony Space display with notes labelled using Roman numerals. The roots of an Aeolian VI II V I progression are shown.

In Harmony Space, direct manipulation theory (Hutchins, Hollans & Norman, 1986) is applied to these cognitive models to produce interactive interfaces in which notes, intervals, chords, chord sequences and modulations can be directly manipulated and visualised, using a single spatial metaphor, via a two handed direct manipulation interface. Interestingly, demands arising purely from the theory of Direct Manipulation and decisions about which aspects of Harmony to teach, lead to several variants of Longuet-Higgins' theory, the last of which is mathematically equivalent to Balzano's theory, given a suitable metric on the space. Given that the two theories start from quite different characterisations of pitch, and emphasise different properties, this is a surprising finding.

The tool is analogous to interactive scientific visualisation tools, except that instead of being designed for the direct visualisation of a physical phenomenon, it is used for the interactive control of objects and relationships as represented in an abstract, cognitive theory. Harmony Space is generally applicable to tonal harmony and to some microtonal pitch systems. The most developed version, based on Balzano's theory of harmony, has some features that work very naturally with jazz and popular harmony. However, the tool can be applied equally effectively to work on conventional functional harmony, and has been used by Howard (Howard, Holland & Whitelock, 1994) for teaching harmonisation of Bach chorales and simple harmonic analysis of Mozart pieces.

Harmony Space is a family of tools rather than a single tool. Variants of the interface focus variously on harmonisation and composition, harmonic analysis, microtonal pitch spaces, studying compositional implications of just temperament, and interfacing with the cognitive support framework MC described in the last section. Other uses include teaching aspects of music theory and music fundamentals. Simple versions of the tool have been shown to be well-suited for practical one to one teaching , and teaching in small groups (Whitelock, Holland & Howard, 1994, 1995), but it is clear that students need guidance from a teacher or courseware to use the program effectively.

Harmony Space makes it relatively easy to grasp a range of harmonic relationships that are not apparent to beginners, and some relationships that are not apparent to trained musicians. It makes relatively sophisticated harmonic strategies accessible to the beginning learner. The system is firmly based on AI representations and theories, but no knowledge-intensive inference techniques are required at the point of delivery. The system is coded more or less conventionally using object-oriented techniques. However, the methodology and sensibility behind Harmony Space are strongly based in AI.

The final section of the paper looks at recent sophisticated techniques for allowing ILEs and student to work together co-operatively. There are many situations in which it would be wrong to assume that a music tutoring system necessarily knows better than the student. In such situations there is a need to focus squarely on issues other than the transmission of facts and techniques to the student. In particular, the next section focuses on negotiation, dialogue and reflection.

## 8. Teaching by Negotiation, Dialogue and Reflection

### 8.1 Grouping and expressive performance: teaching by negotiation

Baker's system (1990) takes a different approach from any of the previous systems to the problem of teaching in an area where the tutor's knowledge is likely to be incomplete, and where the learner's knowledge may, in some cases, be better than the tutor's. The area used as a vehicle to explore this question is the expressive performance of tonal music. Using a computer to assist in teaching musical expression has interesting possibilities, in that even students who cannot yet play an instrument may be given the opportunity to explore and create contrasting expressive performances in a principled way. Expressive performance is taken as covering principally rubato, fluctuations in dynamics, and variations in articulation.

Various researchers, including Sundberg et al. (1989), Todd (1989) and Clynes (1983) have looked at the problem of trying to specify explicitly 'musically appropriate' performances from a score. Sundberg's theory, for example, specifies various simple actions, such as inserting a pause in between two notes, or shortening a note, in response to features found in the melody. Typical such features include melodic leaps and ascending runs. This approach focuses mostly on musical surface features, rather than on large scale harmonic structure. By contrast, Todd's model uses parabolic curves based on a hierarchical phrase structuring to quantitatively predict expressive timing. Clynes proposes composer-specific and metre-specific tempo patterns.

Baker's work assumes that the way in which a performer perceives the grouping structure of a piece is an important influence on their expressive performance. Expert listeners may disagree about the preferred grouping for a given piece or fragment (for example, the first two bars of Mozart Piano Sonata K331), but they do tend to agree whether a given grouping is plausible or not. However, current AI theories of phrase grouping are currently extremely imperfect. Hence AI systems sometimes produce groupings that few, if any, expert listeners would find plausible.

This limitation poses a problem for any AI-ED system designed to advise a student on the expressive performance of a piece. Baker's approach to this problem of uncertainty is to devise mechanisms and representations for the system and the student to negotiate with each other. Once such mechanisms are found, there are many ways they might be applied to the teaching process. In general, the system and the tutor might negotiate about what examples to look at, what teaching strategy to use, what to do next, and what opinions to accept or reject, etc. Such a system is said to support 'learning by dialogue'. The basic actions of such a system include making claims and giving evidence for claims. If challenged by the user with good supporting evidence, the system should be prepared to retract a claim and proceed accordingly. Similarly, the system should be able to rebut claims by a student where her supporting evidence is poor, and perhaps make counter-claims. This technique is generally useful in many domains other than music. Baker's system is primarily intended as a system for exploring the theory of teaching by negotiation, as opposed to a fully developed practical system for the class-room. For detailed information, see Baker (1989, 1990, 1994).

### 8.2 Supporting reflection in music composition

Cook's work (Cook, 1994; Cook & Morgan, 1995), like Baker's work, focuses on learning through dialogue. In Cook's case there is a strong explicit emphasis on finding a framework for describing both learners' and teachers' internal dialogues at several levels. The central aim is to provide a theoretical framework for describing learning and teaching processes in music composition. Two applications of the framework include a prototype Interactive Learning Environment and a method for analysing protocols with the aim of understanding teacher-student interaction. The term 'protocol' as used in AI-ED typically refers to transcriptions of dialogues between teachers and students, or sometimes between two or more students. Protocol analysis is seen in AI-ED as a useful tool for studying learning and teaching processes.

Cook's system COLERIDGE (Composition Learning Environment For Reflection about Intentions and Dialogue Goals in Education) makes no attempt to compose itself, since the main thrust is to model higher level creative activities. However, it does provide various music transformation tools, similar to those found in many Music Logo systems. Indeed, the current version of COLERIDGE is implemented in Peter Stones' Symbolic Composer (Morgan and Tolonen, 1995), a version of Lisp with a wide-ranging music function library.

Cook's system is not designed to communicate musical knowledge as such, but to foster higher level skills, in particular, skills that support *reflection* and *problem seeking*. Reflection, according to John Dewey (1916), the American pragmatic philosopher and educator, is the "intentional endeavour to discover specific connections between something which we do and the consequences which result." In effect, reflection is an aim of striving continually to learn and re-think one's activities in the light of the outcomes of one's actions, seen in their contexts. The notion of reflection has been a strong influence in disciplines such as Design (Schön, 1993) and Education (Carr, 1989). The concept of reflection as developed by Lipman (1991a) is used to refer to situations in which students are encouraged not just to seek out new problems (problem seeking), but also to go beyond the immediate problem and solution to improve their ability to find solutions. Both the idea of reflection and the idea of problem seeking are highly relevant to music education. A good example of reflection and problem seeking in music would be a composer who, when composing, monitors her own activities and tries to make generalisations from them. Thus, instead of merely applying known compositional methods and techniques, the reflective composer develops and refines new compositional methods and techniques, and strives to develop her ability to create new methods. Similarly, the reflective teacher, while teaching, will try to improve on her teaching, and the reflective learner will be looking out pro-actively for new ways of learning.

In order to model reflective teaching and learning activities, Cook's framework models both teacher and learner in two different roles; namely the teacher both as a composer and as a teacher, and the learner both as a composer and as a learner. Within each of these roles, four levels are provided, with each higher level reflecting on the activities of the levels below. The precise distinction between each level is still undergoing refinement. Cook's prototype system COLERIDGE is currently focused around tasks such as transforming motivic material, say for a piano study piece, and reflecting on the results. The work is still in progress and at present it remains to be seen the extent to which the theoretical framework will lead to a detailed implementation. From an AI and music point of view, the idea of being explicit about reflection and applying it to music is novel and interesting. From an AI-ED point of view, the development and application of ideas of reflection in a problem seeking domain makes the work of great interest.

## 9. Summary and conclusions

Music is an open-ended area in which there are few pre-defined goals or rules, but in which there are pronounced differences between beginners and experts. In such a domain, problem seeking is at least as important as problem solving.There are a variety of ways of applying AI to Music Education. Applications exist in areas such as ear training, music theory, listening, composition and performance. In this paper we have focused mostly on composition, but with some attention to performance and listening. In the areas for which the Logo approach is best known (Mathematics, Physics, etc.), claims about its effectiveness are disputed. The approach has strong adherents, but critics claim that highly motivated and informed educators are required to make it work, and that it is unclear where to assign credit for successes (i.e. to the system or to the teacher). In open ended areas such as music composition, there is a much stronger consensus that the Logo approach is valuable, though this has never been investigated in depth empirically. Many opportunities exist to extend Music Logo work to different compositional techniques and to other aspects of music. Intelligent Tutoring Systems are well suited, in general, to areas in which there are hard and fast rules and goals, or ways of identifying and categorising systematic errors. Such areas are in short supply in music. Some obvious application areas such as four part harmony and counterpoint have been investigated. There is scope for refining existing work in such areas, and for identifying other relatively well-defined areas in music where hard and fast rules and goals apply. For example, there may be applications in ear training. However it is important to realise the limitations of the technique, and to refrain from applying it where it is inappropriate.

Harmony Space is an example of a Human-Interface-centred approach that draws on AI theories of music and AI methodologies. This system has various sources of power. One applies very widely both inside and outside music: Harmony Space draws on AI theories of a domain in music (harmony) and uses them in place of a task model in the human computer interface. Direct Manipulation techniques are then applied, modifying the domain theory where necessary, to render perceptually salient those musical materials and relationships that are deemed conceptually important. This technique is very widely applicable. Another source of power comes from the domain theories used, especially Balzano's (1980) theory. An interesting research project would be to explore whether similar power in an interface could be gained from group theoretic characterisations of other domains, not necessarily connected with music.

The MC Cognitive Support Framework is applicable in any open-ended problem seeking area, not just music. It gains particular power in the case of harmony for a variety of reasons. These include: employing the elegant representations for harmony used in Harmony Space; generalising over tonal and modal properties; and using generative plans of a kind that afford multiple viewpoints. Similar ideas could be applied in other parts of music, as demonstrated by MOTIVE.

Negotiation is an important open research area of AI and AI-ED, applicable to many domains. In the case of music education, there are applications for dealing with the limitations and incompleteness of AI theories of music, and for exploring human-machine co-operation. Given the current amounts of domain knowledge that such systems typically possess, and given direct manipulation and visualisation techniques available for facilitating human machine communication, it is less clear the extent to which this work is practically useful in presently deliverable systems for music education.

Reflection is a valuable principle for encouraging problem seeking behaviour, and for designing systems that support problem seeking. The potential has been demonstrated, but much work remains in devising ways of applying reflection in practicable deliverable systems.

## 10. References


Ames, C. (1989). The Markov Process as a Compositional Model: A Survey and Tutorial. Leonardo, 22(2), 175-187.

Baker, M. (1989). An artificial intelligence approach to musical grouping analysis. Contemporary Music Review, 2, 43-68.

Baker, M. (1990). Arguing with the Tutor. In M. Elsom-Cook (Eds.), Guided Discovery Tutoring London: Paul Chapman Publishing.

Baker, M. (1994). A Model for Negotiation in Learning Teaching Dialogues. Journal of Artificial Intelligence in Education, 5(2).

Balzano (1980). The Group-theoretic Description of 12-fold and Microtonal Pitch systems. Computer Music Journal, 4(4).

Bamberger, J. (1986). Music Logo. Cambridge, Mass. Terrapin Inc.

Bamberger, J. (1991). The Mind Behind the Musical Ear. Cambridge, Mass. Harvard University Press.

Binary Designs (1996). Band In A Box.

Camurri, A.,Canepa, C., Frixione, M. & Zaccaria, R. (1991). HARP: A System for Intelligent Composer's Assistance. IEEE Computer, 24(7), 64-67.

Carbonell, J. R. (1970). AI in CAI: an artificial intelligence approach to computer-aided instruction. IEEE Transactions on Man-Machine Systems(MMS), 11(4), 190-202.

Carr, W. (1989). Quality in Teaching. Brighton: Falmer Press.

Clynes, M. (1983). Expressive Microstructure in Music, liked to Living Qualities. In J. Sundberg (Ed.), Studies of Music Performance Stockholm: Royal Swedish Academy of Music.

Cook, J. (1994). Agent Reflection in an Intelligent Learning Environment Architecture for Composition. In M. Smith, A. Smaill, & G. A. Wiggins (Eds.), Music Education: An Artificial Intelligence Approach London: Springer Verlag.

Cook, J. and Morgan, N.(1995). COLERIDGE: Composition Learning Environment for Reflection about Intentions and Dialogue Goals in Education. In A. Smaill (Ed.), International Congress in Music and Artificial Intelligence, University of Edinburgh, Department of Music.

Cork, C. (1988). Harmony by LEGO bricks.

Cumming, N. (1992). Eugene Narmour's Theory of Melody. Music Analysis, 11(203), 354-374.

Desain, P., & Honing, H. (1986). LOCO, composition microworlds in Logo. In P.Berg (Ed.), International Computer Music Conference, (pp. 109-118). San Francisco: Computer Music Association.

Desain, P., & Honing, H. (1992). Music, Mind and Machine. Studies in Computer Music, Music Cognition and Artificial Intelligence. Amsterdam: Thesis Publishers.

Desain, P., & Honing, H. (1996). LOCO-SONNET: a graphical dataflow language for algorithmic composition.

Dewey, J. (1916). Democracy and Education. The Macmillan Company.

Elsom-Cook, M. (Ed.). (1990). Guided Discovery Tutoring. London: Paul Chapman Publishing.

Fenton, A. (1989). The Design of an Intelligent Tutoring System for Music. Musicus, 1(2), 125-143.

Feynman, R. P. (1986). Surely you're joking Mr Feynman. London: Unwin.
Fux, J. J. (1725/1965). Gradus ad Parnassum. New York, Norton.
Gross, D. (1984). Computer applications to Music theory: a retrospective. Computer Music Journal, 8(4).
Hofstetter, F. (1981). Computer-Based Aural Training: the Guido System. Journal of Computer-Based Instruction 7(3): 84-92.
Holland, S. (1987). A knowledge-based tutor for music composition. (OU IET CITE Report No. 16, June 1987. No. 16).
Holland, S. (1989) Artificial Intelligence, Education and Music. PhD, Open University.
Holland, S. (1992). Interface design for empowerment: a case study from music. Multimedia Interface Design in Education. A. D. N. Edwards and S. Holland. Berlin, Springer Verlag.
Holland, S. (1991). Preliminary report on the design of a constraint-based musical planner (Technical Report No. AUCS/TR9113). Department of Computing Science, Kings College, University of Aberdeen.
Holland, S. (1994). Learning about harmony with Harmony Space: an overview. In M. Smith, Smaill.A, & Wiggins.G. (Eds.), Music Education: an artificial Intelligence approach London: Springer Verlag.
Holland, S., & Elsom-Cook, M. (1990). Architecture of a knowledge-based music tutor. In M. Elsom-Cook (Eds.), Guided Discovery Tutoring London: Paul Chapman Publishing.
Honing, H. (1990). POCO: an environment for analysing, modifying, and generating expression in music. In Proceedings of the 1990 International Computer Music Conference, (pp. 364-368). San Francisco: Computer Music Association.
Honing, H. (1992). Expresso, a strong and small editor for expression. In Proceedings of the 1992 International Computer Music Conference, (pp. 215-218). San Francisco: Computer Music Association.
Howard, P., Holland, S., & Whitelock, D. (1994). Keyboard Harmony: some applications of computers in music education. Musical Times, 467-471.
Hutchins, E. L., Hollans, J. D., & Norman, D. A. (1986). Direct Manipulation Interfaces. In D. A. Norman & S. Draper (Eds.), User Centred System Design: New perspectives on Human Computer Interaction, Hillsdale, NJ.: Erlbaum.
Jameson (1992). Sonnet: Audio-Enhanced Monitoring and Debugging. In The International Conference on Auditory Display '92.
Johnson-Laird, P. (1991). Jazz Improvisation: A Theory at the Computational Level. In P. Howell, R. West, & I. Cross (Eds.), Representing Musical Structure London: Academic Press.
Johnson-Laird, P. N. (1988a). The Computer and the Mind. London: Fontana.
Johnson-Laird, P. N. (1988b). Freedom and Constraint in Creativity. In Sternberg (Ed.), The Nature of Creativity (pp. 202-219). Cambridge University Press.
Kane, T. (1991) Another Prototype implementation of the Rhythm Machine. Masters Thesis, University of Aberdeen, Department of Computing Science.
Kommers, P., Jonassen, D., & Mayes, T. (1992). Cognitive Tools for Learning. Berlin: Springer.
Laurillard, D. (1993). Rethinking university teaching : a framework for the effective use of educational technology. London: Routledge.
Lenat, D. B., & Brown, J. S. (. (1984). Why AM and Eurisko appear to work. Artificial Intelligence, 21, 31-59.
Lerdahl, F., & Jackendoff, R. (1983). A Generative Theory of Tonal Music. London: MIT Press.
Levitt (1986). Hookup. In Proceedings of the Conference on Small Computers and the Arts, Philadelphia.
Levitt, D. A. (1981) A Melody Description System for Jazz Improvisation. MS Thesis, Massachusetts Institute of Technology.
Levitt, D. A. (1985) A Representation for Musical Dialects. PhD, Massachusetts Institute of Technology.
Lipman (1991a). Thinking in Education. Cambridge: Cambridge University Press.
Lipman, M. (1991b). Strengthening reasoning and judgement through philosophy. In S.Maclure & P.Davies (Ed.), Learning to think: thinking to learn. Proceedings of the 1989 OED conference,. Persimmon Press.
Longuet-Higgins, H. C. (1962). Letter to a Musical Friend. Music Review(August 1962), 244-248.
McAdams, S. and A. Bregman (1985). Hearing Musical Streams. The Foundations of Computer Music. C. Roads and J. Strawn. Boston, MIT Press.
Mehegan, J. (1959). Jazz Improvisation I: Tonal and Rhythmic Principles. New York. Watson-Guptil Publications.
Menzel, E. W. (1991). Chimpanzees (Pan troglodytes): Problem seeking versus the bird-in-hand, least-effort strategy. Primates, 32, 497-508.
Minsky, M. (1981). Music, Mind and Meaning. Computer Music Journal, 5(3), 28-44.

Moore, A. (1992). Patterns of harmony. Popular Music, 11(1), 73-106.
Morgan, N., & Tolonen, P. (1995). Symbolic Composer Professional - a software application for Apple Macintosh computers.
Narmour, E. (1989). The "genetic code" of melody: Cognitive structures generated by the implication-realisation model. Contemporary Music Review, 4, 45-63.
Naughton, J. (1986). Artificial Intelligence and Industrial Training. Open University Systems Group/ Manpower Services Commission.
Newcomb, S. R. (1985). LASSO: An intelligent Computer Based Tutorial in Sixteenth Century Counterpoint. Computer Music Journal, 9(4).
O'Shea, T., & Self, J. (1983). Learning and Teaching with Computers. London: Prentice-Hall.
Pachet, F. (1994). The MusES system : an environment for experimenting with knowledge representation techniques in tonal harmony. In First Brazilian Symposium on Computer Music, SBC&M, (pp. 195-201). Caxambu, Minas Gerais, Brazil.
Papert, S. (1980). Mindstorms. Brighton. The Harvester Press.
Pena, W. (1987). Problem seeking : an architectural programming primer (3rd ed.). Washington: AIA Press.
Pierce, J. R. (1983). The Science of Musical Sound, Scientific American Books, New York.
Pratt, G. (1984). The Dynamics of Harmony. Milton Keynes, Open University Press.
Puckette, M. (1988). The Patcher. In Proceedings of the International Computer Music conference 1988. Feedback papers 33, Feedback Studio Verlag, Cologne.
Rittel, H. W. J., & Webber, M. W. (1984). Planning Problems are wicked problems. In N. Cross (Eds.), Developments in Design Methodology Chichester: John Wiley and Sons.
Roads, C. (1996). The Computer Music Tutorial. MIT Press.
Sanchez, M., Joseph, A., Dannenberg, R., Miller, P., Capell, P., & Joseph, R. (1987). Piano tutor: An Intelligent Keyboard Instruction System. In Proceedings of the First International Workshop on Artificial Intelligence and Music,. St. Paul, Minnesota: AAAI.
Schaffer, J. W. (1991). Harmony Coach: An Exploration of Microcomputer-Based Intelligent Tutoring Systems in Music. Journal of Computer Based Instruction, 18(1), 30-36.
Schon, D. A. (1993). The Reflective Practitioner. New York: Basic Books.
Self, J. (1995). Computational Mathetics: Towards a Science of Learning Systems Design (CBLU Report No. 96/24). Computer Based Learning Unit, University of Leeds.
Sharples, M. (1983). The use of Computers to Aid the Teaching of Creative Writing. AEDS Journal.
Sleeman, D., & Brown, J. S. (Ed.). (1982). Intelligent Tutoring Systems. London: Academic Press.
Sloboda, J. A. (1985). The Musical Mind: The Cognitive Psychology of Music. Oxford: Clarendon Press.
Smith, M., & Holland, S. (1994). Motive: The development of an AI tool for Beginning Melody Composers. In Music Education: an artificial Intelligence approach London: Springer Verlag.
Sorisio, L. ( 1987). Designing an Intelligent Tutoring System in Harmony. In Proceedings of International Computer Music Conference, Computer Music Association, San Francisco.
Spensley, F., & Elsom-Cook, M. (1988). Dominie: Teaching and assessment strategies. (CAL Research Group Technical Report No. 74). Open University, Milton Keynes, Great Britain.
Steedman, M. (1972) The Formal Description of Musical Perception. PhD, University of Edinburgh.
Steedman, M. (1984). A generative grammar for chord sequences. Music Psychology, 2(1).
Sundberg, J., A., Friberg, & Fryden, L. (1989). Rules for Automated Performance of Ensemble Music. Contemporary Music Review, 3.
Taube, H. (1991). Common Music: A Music Composition Language in Common Lisp and CLOS. Computer Music Journal, 15(2), 21-32.
Thomas, M. T. (1985). Vivace: A Rule Based AI System for Composition. In International Computer Music Conference, (pp. 267-274). San Francisco.
Todd, N. (1989). A Computational Model of Rubato. In E. Clarke & S. Emmerson (Eds.), Music, Mind and Structure
Van Hentenryk, P., & Dincbas (1987). Forward checking in Logic Programming. In Proceedings of Fourth International Conference on Logic programming, (pp. 229-256). MIT Press.
Voyager (1989). Beethoven's Ninth Symphony Interactive CD ROM.
Watson, C. (1990) A Prototype implementation of the Rhythm Machine. Masters Thesis, University of Aberdeen, Department of Computing Science.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos, California. Morgan Kaufmann Publishers Inc.

Whitelock, D., Holland, S., & Howard, P. (1994). Groupwork, Computers and Music Education: an Evaluation. In *Proceedings of the Group and Interactive Learning Conference*, Strathclyde.

Whitelock, D., Holland, S., & Howard, P. (1995). Understanding Harmony and Technology in Music Education. In T. Sechrest, M. Thomas, & N.Estes (Ed.), *Proceedings of the 12th International Conference on Technology and Education*, (pp. 382-384). Orlando, Florida,: University of Texas.

Yavelow, C. (1992). *MacWorld Music & Sound Bible.* San Mateo, California. IDG Books.